UNIVERSITI SAINS MALAYSIA

Second Semester Examination
*[Peperiksaan Semester Kedua]*

Academic Session 2007/2008
*[Sidang Akademik 2007/2008]*

April 2008

# CST331/CST315 – Principles of Parallel & Distributed Programming
## *[Prinsip-Prinsip Pengaturcaraan Selari & Teragih]*

Duration : 2 hours
*[Masa : 2 jam]*

---

**INSTRUCTIONS TO CANDIDATE:**
***[ARAHAN KEPADA CALON:]***

- Please ensure that this examination paper contains **FOUR** questions in **NINE** printed pages before you begin the examination.

  *[Sila pastikan bahawa kertas peperiksaan ini mengandungi **EMPAT** soalan di dalam **SEMBILAN** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

- Answer **ALL** questions.

  *[Jawab **SEMUA** soalan.]*

- You may answer the questions either in English or in Bahasa Malaysia.

  *[Anda dibenarkan menjawab soalan sama ada dalam Bahasa Inggeris atau Bahasa Malaysia.]*

---

1. (a) Software engineering support is said to be even more important for parallel computing than sequential computing. Do you agree with this statement? Provide **two (2)** reasons to support your answer.

(5/100)

(b) State **two (2)** metrics that are commonly used for:

(i) single-application performance.

(ii) multi-application performance.

Briefly explain each of the metrics.

(10/100)

(c) What is a cluster? Why a cluster architecture machine is gaining popularity? Give the architecture specifications of the *inderapura* cluster that you have used, in terms of the following:

(i) number of nodes.

(ii) processor type.

(iii) topology.

(10/100)

2. (a) (i) A program is to sum two vectors to produce a third vector. The vector elements are stored in two arrays, A and B, and we need to produce a third vector C. The basic loop is:

```
for (i = 0; i < N; i++)
    C[i] = A[i] + B[i];
```

We can parallelise the program above by partitioning the N elements into P partitions of N/P elements each and assign to respective processors. This approach is called "loop splitting". Suggest **two (2)** ways of loop splitting the loop above. (You can use example to clarify your answer).

(5/100)

(ii) Parallelise the loop above using **one (1)** of the methods given in your answer in Question 2(a)(i) using

- OpenMP directives
- PThread libraries

(10/100)

(b) (i) Discuss the **four (4)** phases of parallel algorithm design in the context of shared memory programming model. Relate with either a crowd simulation or a string matching application.

(6/100)

(ii) Can the Diffusion algorithm be used in balancing the workload in question 2(b)(i)? Why or why not?

(4/100)

3. (a) (i) Modeling the performance of a parallel algorithm is an important aspect in designing a parallel solution for an application. Why is performance modeling important?

(3/100)

(ii) There are two approaches in performance modeling. Name the **two (2)** approaches and briefly discuss the disadvantage of each.

(7/100)

(b) (i) Name **four (4)** groups of MPI library calls and give **one (1)** example for each.

(4/100)

(ii) SUM is a program that will add a list of numbers. Below is the skeleton of a C++ program. Fill in section (A), (B), (C), (D) and (E) with the correct C or C++ statements and MPI library routines.

```
// SUM is an example program which sum up a list of numbers
// kept in an array using the MPI message passing
// interface library.

// include all header files

# define N 100

int main ( int argc, char *argv[] )
{

   double array[N], seed, PI = 3.14159265358979323846264643,
         sum, sum_all;
   int i, master = 0, my_id, num_procs, tag;
   MPI::Status status;

   // Section(A)- Initialize MPI.

   num_procs = MPI::COMM_WORLD.Get_size ( );

   // Section (B)- Determine the rank of this process.

   if ( my_id == master ){
      // master prints information
   }

   if ( my_id == master ){
      // The master process initialize by reading input
      // into the array.
   }

   // Section (C)- The master process broadcasts the
   // computed initial values to all the other processes.

   sum = 0.0;
   for ( i = 0; i < N; i++ ){
      sum = sum + array[i] * ( double ) my_id;
   }

   cout << "SUM - Process " << my_id << ":\n";

   // Section (D)- Each worker process sends its sum
   // back to the master process.

   // Section (E) - Terminate MPI

   if ( my_id == master ){
      // master print result
   }
   return 0;
}
```

(11/100)

4. (a) One of the communication mechanisms of the client-server model is the Remote Procedure Call (RPC).

(i) How is RPC differs from the normal procedure call?

(3/100)

(ii) A stub is a local procedure that manages communication on behalf of its client or server. Describe detail activities of stub when an RPC is invoked. Use a diagram to assist your explanation.

(7/100)

(iii) State **one (1)** advantage and **one (1)** disadvantage of RPC compared to sockets.

(5/100)

(b) (i) The actor model is an example of a distributed object models. Describe what is actor model.

(4/100)

(ii) The model supports distributed as well as parallel computing. State **two (2)** forms of parallelism which can be exploited using the model.

(6/100)

1. (a) Sokongan kejuruteraan perisian dikatakan lebih diperlukan bagi perkomputeran selari berbanding perkomputeran berjujukan. Adakah anda setuju dengan penyataan ini? Berikan **dua (2)** sebab bagi menyokong jawapan anda.

(5/100)

(b) Nyatakan **dua (2)** metrik yang biasanya digunakan bagi:

(i) prestasi aplikasi tunggal.

(ii) prestasi aplikasi berganda.

Terangkan dengan ringkas setiap metrik.

(10/100)

(c) Apakah itu kluster? Kenapa mesin seni bina kluster semakin mendapat sambutan? Berikan spesifikasi seni bina kluster *inderapura* yang anda telah guna dari sudut:

(i) bilangan nod.

(ii) jenis pemproses.

(iii) topologi.

(10/100)

2. (a) (i) Satu atur cara menjumlahkan dua vektor dan menghasilkan vektor ketiga. Semua elemen-elemen dalam vektor disimpan di dalam dua tatasusunan berasingan A dan B dan hasil disimpan di dalam vektor ketiga C. Gelung asas untuk atur cara tersebut ialah:

```
for (i = 0; i < N; i++)
    C[i] = A[i] + B[i];
```

Kita boleh menselarikan atur cara di atas dengan membahagikan N elemen kepada P bahagian di mana setiap bahagian adalah bersaiz N/P dan tugaskan kepada p pemproses. Kaedah ini dinamakan "pemecahan gelung". Cadangkan **dua (2)** kaedah bagaimana pemecahan gelung ini boleh dilakukan. (Anda boleh menggunakan contoh untuk menjelaskan jawapan anda).

(5/100)

(ii) Selarikan gelung di atas dengan menggunakan salah **satu (1)** kaedah yang anda beri dalam jawapan dalam Soalan 2(a)(i) di atas menggunakan

- arahan OpenMP
- perpustakaan PThreads

(10/100)

(b) (i) Bincang **empat (4)** fasa reka bentuk algoritma selari di dalam kontex model pengaturcaraan ingatan terkongsi. Hubungkan dengan aplikasi simulasi orang ramai atau pemadanan rentetan.

(6/100)

(ii) Bolehkah algoritma Diffusion digunakan untuk mengimbangkan beban kerja di soalan 2(b)(i)? Kenapa atau kenapa tidak?

(4/100)

3. (a) (i) Pemodelan prestasi algoritma selari merupakan aspek penting dalam mereka bentuk penyelesaian selari bagi sesuatu aplikasi. Kenapakah pemodelan prestasi itu penting?

(3/100)

(ii) Terdapat dua pendekatan dalam pemodelan prestasi. Namakan **kedua-dua (2)** pendekatan tersebut dan bincangkan dengan ringkas kelemahan setiap satu darinya.

(7/100)

(b) (i) Namakan **empat (4)** kumpulan panggilan perpustakaan MPI dan beri **satu (1)** contoh yang sesuai untuk setiap satu.

(4/100)

(ii) SUM adalah satu atur cara yang menjumlahkan satu senarai nombor-nombor. Diberi di bawah rangka bagi atur cara dalam C++. Isikan bahagian yang bertanda (A), (B), (C), (D) dan (E) dengan penyataan C atau C++ dan rutin perpustakaan MPI yang betul.

```
// SUM is an example program which sum up a list of numbers
// kept in an array using the MPI message passing
// interface library.

// include all header files

# define N 100

int main ( int argc, char *argv[] )
{

   double array[N], seed, PI = 3.14159265358979323846264 3,
          sum, sum_all;
   int i, master = 0, my_id, num_procs, tag;
   MPI::Status status;

   // Section(A)- Initialize MPI.

   num_procs = MPI::COMM_WORLD.Get_size ( );

   // Section (B)- Determine the rank of this process.

   if ( my_id == master ){
      // master prints information
   }

   if ( my_id == master ){
      // The master process initialize by reading input
      // into the array.
   }

   // Section (C)- The master process broadcasts the
   // computed initial values to all the other processes.

   sum = 0.0;
   for ( i = 0; i < N; i++ ){
      sum = sum + array[i] * ( double ) my_id;
   }

   cout << "SUM - Process " << my_id << ":\n";

   // Section (D)- Each worker process sends its sum
   // back to the master process.

   // Section (E) - Terminate MPI

   if ( my_id == master ){
      // master print result
   }
   return 0;
}
```

(11/100)

4. (a) Salah satu mekanisma komunikasi model pelanggan-pelayan ialah Panggilan Prosedur Terpencil.

(i) Bagaimanakah Panggilan Prosedur Terpencil berbeza dari panggilan prosedur biasa?

(3/100)

(ii) Suatu "stub" ialah prosedur setempat yang menguruskan komunikasi bagi pihak pelanggan atau pelayannya. Huraikan suatu aktiviti "stub" dengan terperinci apabila Panggilan Prosedur Terpencil dipanggil.

(7/100)

(iii) Nyatakan **satu (1)** kelebihan dan **satu (1)** kekurangan Panggilan Prosedur Terpencil berbanding dengan soket.

(5/100)

(b) (i) Model "actor" merupakan suatu contoh model objek teragih. Beri huraian model "actor".

(4/100)

(ii) Model berkenaan menyokong perkomputeran teragih dan juga selari. Bincangkan dua bentuk penselarian yang boleh dieksploitasi menggunakan model tersebut.

(6/100)

- oooOooo -